THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Greatest Fixed Points of Probabilistic Min/Max Polynomial Equations, and Reachability for Branching Markov Decision Processes?

Citation for published version:
Etessami, K, Stewart, A & Yannakakis, M 2015, Greatest Fixed Points of Probabilistic Min/Max Polynomial Equations, and Reachability for Branching Markov Decision Processes? in *Automata, Languages, and Programming: 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II.* Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-47666-6_15

Digital Object Identifier (DOI):
10.1007/978-3-662-47666-6_15

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Automata, Languages, and Programming

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 05. May. 2023

Greatest Fixed Points of Probabilistic Min/Max Polynomial Equations, and Reachability for Branching Markov Decision Processes*

Kousha Etessami[1], Alistair Stewart[1], and Mihalis Yannakakis[2]

[1] School of Informatics, University of Edinburgh
kousha@inf.ed.ac.uk , stewart.al@gmail.com
[2] Department of Computer Science, Columbia University
mihalis@cs.columbia.edu

Abstract. We give polynomial time algorithms for quantitative (and qualitative) *reachability* analysis for *Branching Markov Decision Processes* (BMDPs). Specifically, given a BMDP, and given an initial population, where the objective of the controller is to maximize (or minimize) the probability of eventually reaching a population that contains an object of a desired (or undesired) type, we give algorithms for approximating the supremum (infimum) reachability probability, within desired precision $\epsilon > 0$, in time polynomial in the encoding size of the BMDP and in $\log(1/\epsilon)$. We furthermore give P-time algorithms for computing ϵ-optimal strategies for both maximization and minimization of reachability probabilities. We also give P-time algorithms for all associated *qualitative* analysis problems, namely: deciding whether the optimal (supremum or infimum) reachability probabilities are 0 or 1. Prior to this paper, approximation of optimal reachability probabilities for BMDPs was not even known to be decidable.
Our algorithms exploit the following basic fact: we show that for any BMDP, its maximum (minimum) *non*-reachability probabilities are given by the *greatest fixed point* (GFP) solution $g^* \in [0,1]^n$ of a corresponding monotone max (min) Probabilistic Polynomial System of equations (max/min-PPS), $x = P(x)$, which are the Bellman optimality equations for a BMDP with non-reachability objectives. We show how to compute the GFP of max/min PPSs to desired precision in P-time.

1 Introduction

Multi-type branching processes (BPs) are infinite-state purely stochastic processes that model the stochastic evolution of a population of entities of distinct types. The BP specifies for every type a probability distribution for the offspring of entities of this type. Starting from an initial population, the process evolves from each generation to the next according to the probabilistic offspring

* The full version of this paper is available at arxiv.org/abs/1502.05533. Research partially supported by the Royal Society and by NSF Grant CCF-1320654. Alistair Stewart's research supported by I. Diakonikolas's EPSRC grant EP/L021749/1.

rules. Branching processes are a fundamental stochastic model with applications in many areas: physics, biology, population genetics, medicine etc. *Branching Markov Decision Processes* (BMDPs) provide a natural extension of BPs where the evolution is not purely stochastic but can be partially influenced or controlled: a controller can take actions which affect the probability distribution for the set of offspring of the entities of each type. The goal is to design a policy for choosing the actions in order to optimize a desired objective.

In recent years there has been great progress in resolving algorithmic problems for BMDPs with the objective of maximizing or minimizing the *extinction* probability, i.e., the probability that the population eventually becomes extinct. Polynomial time algorithms were developed for both maximizing and minimizing BMDPs for *qualitative* analysis, i.e. to determine whether the optimal extinction probability is 0, 1 or in-between [12], and for *quantitative* analysis, to compute optimal extinction probabilities to any desired precision [9]. However, key problems for optimizing BMDP *reachability* probability (probability that the population eventually includes an entity with a target type) have remained open.

Reachability objectives are very natural. Some types may be undesirable, in which case we want to avoid them to the extent possible. Or conversely, we may want to guide the process to reach certain desirable types. For example, branching processes have been used recently to model cancer tumor progression and multiple drug resistance of tumors due to multiple mutations ([1, 15]). It could be fruitful to model the introduction of multiple drugs (each of which controls/influences cells with a different type of mutation) via a "controller" that controls the offspring of different types, thus extending the current models (and associated software tools) which are based on BPs only, to controlled models based on BMDPs. A natural question one could ask then is to compute the minimum probability of reaching a *bad* (malignant) cell type, and compute a drug introduction strategy that achieves (approximately) minimum probability. Doing this efficiently (in P-time) would avoid the combinatorial explosion of trying all possible combinations of drug therapies.

In this paper we provide the first polynomial time algorithms for quantitative (and also qualitative) *reachability* analysis for BMDPs. Specifically, we provide algorithms for ϵ-approximating the supremum probability, as well as the infimum probability, of reaching a given type (or a set of types) starting from an initial type (or an initial population of types), up to any desired additive error $\epsilon > 0$. We also give algorithms for computing ϵ-optimal strategies which achieve such ϵ-optimal values. The running time of these algorithms (in the standard Turing model of computation) is polynomial in both the encoding size of the BMDP and in $\log(\frac{1}{\epsilon})$. We also give P-time algorithms for the qualitative problems: we determine whether the supremum or infimum probability is 1 (or 0), and if so we actually compute an optimal strategy that achieves 1 (0, respectively).

In prior work [12], we studied optimization of extinction (a.k.a. termination) probabilities for BMDPs, and showed that optimal extinction probabilities are captured by the *least fixed point* (LFP) solution $q^* \in [0,1]^n$ of a corresponding system of monotone probabilistic max (min) polynomial equations called

maxPPSs (respectively minPPSs), which form the *Bellman optimality equations* for termination of a BMDP. A maxPPS is a system of equations $x = P(x)$ over a vector x of variables, where the right-hand-side of each equation is of the form $\max_j\{p_j(x)\}$, where each $p_j(x)$ is a polynomial with non-negative coefficients (including the constant term) that sum to at most 1 (such a polynomial is called *probabilistic*). A minPPS is defined similarly. In [9], we introduced an algorithm, called *Generalized Newton's Method* (GNM), for the solution of maxPPSs and minPPSs, and showed that it computes the LFP of maxPPSs and minPPSs (and hence also the optimal termination probabilities for BMDPs) to desired precision in P-time. GNM is an iterative algorithm (like Newton's) which in each iteration solves a suitable linear program (a different one for the max and min versions).

In this paper we first model the reachability problem for a BMDP by an appropriate system of equations: We show that the optimal *non-reachability* probabilities for a given BMDP are captured by the *greatest fixed point* (GFP), $g^* \in [0,1]^n$ of a corresponding maxPPS (or minPPS) system of Bellman equations. We then show that one can approximate the GFP solution $g^* \in [0,1]^n$ of a maxPPS (or minPPS), $x = P(x)$, in time polynomial in both the encoding size $|P|$ of the system of equations and in $\log(1/\epsilon)$, where $\epsilon > 0$ is the desired additive error bound of the solution. (The model of computation is the standard Turing machine model.) We also show that the qualitative analysis of determining the coordinates of the GFP that are 0 and 1, can be done in P-time (and hence the same holds for the optimal reachability probabilities of BMDPs).

Our algorithms for computing the GFP of minPPS and maxPPS make use of (a variant of) Generalized Newton Method adapted for the computation of GFP, with a key important difference in the preprocessing step before applying GNM. We first identify and remove only the variables that have value 1 in the GFP g^* (we do not remove the variables with value 0, unlike the LFP case). We show that for maxPPSs, once these variables are removed, the remaining system with GFP $g^* < 1$ has a unique fixed point in $[0,1]^n$, hence the GFP is equal to the LFP; applying GNM from the 0 initial vector converges quickly (in P-time, with suitable rounding) to the GFP (by [9]). For minPPSs, even after the removal of the variables x_i with $g^*_i = 1$, the remaining system may have multiple fixed points, and we can have LFP $<$ GFP. Nevertheless, we show that with the subtle change in the preprocessing step, GNM, starting at the all-0 vector, remarkably "skips over" the LFP and converges to the GFP solution g^*, in P-time.

Comparing the properties of the LFP and GFP of max/minPPS, we note that one difference for the qualitative problems is that for the GFP, both the value=0 and the value=1 question depend only on the structure of the model and not on its probabilities (the values of the coefficients), whereas in the LFP case the value=1 question depends on the probabilities (see [13, 12]).

We also note some important differences regarding existence of optimal strategies between extinction (termination) and reachability objectives for BMDPs. We observe that, unlike optimization of termination probabilities for BMDPs, for which there always exists a static deterministic optimal strategy ([12]), there need not exist any optimal strategy at all for maximizing reachability probability

in a BMDP, i.e. the supremum probability may not be attainable. If the supremum probability is 1 however, we show that there exists a strategy that achieves it (albeit, not necessarily a static one). For the min reachability objective there always exists an optimal deterministic and static strategy. In all cases, we show that we can compute in P-time an ϵ-optimal static (possibly randomized) policy, for both maximizing and minimizing reachability probability in a BMDP.

Related work: BMDPs have been previously studied in both operations research (e.g., [14, 16]) and computer science (e.g., [12, 6, 11]). We have already mentioned the results in [12, 9] concerning the computation of the extinction probabilities of BMDPs and the computation of the LFP of max/minPPS. BPs are closely connected to stochastic context-free grammars, 1-exit Recursive Markov chains (1-RMC) [13], and the corresponding stateless probabilistic pushdown processes, pBPA [7]; their extinction or termination probabilities are interreducible, and they are all captured by the LFP of PPSs. The same is true for their controlled extensions, for example the extinction probability of BMDPs and the termination probabilities of 1-exit Recursive Markov Decision processes (1-RMDP) [12], are both captured by the LFP of maxPPS or minPPS. A different type of objective of optimizing the total expected reward for 1-RMDPs (and equivalently BMDPs) in a setting with positive rewards was studied in [11]; in this case the optimal values are rational and can be computed exactly in P-time.

The equivalence between BMDPs and 1-RMDPs however does not carry over to the reachability objective. The *qualitative* reachability problem for 1-RMDPs (equivalently BPA MDPs) and the extension to simple 2-person games 1-RSSGs (BPA games) were studied in [4] and [3] by Brazdil et al. It is shown in [4] that qualitative *almost-sure* reachability for 1-RMDPs can be decided in P-time (both for maximizing and minimizing 1-RMDPs). However, for maximizing reachability probability, almost-sure and limit-sure reachability are *not* the same: in other words, the supremum reachability probability can be 1, but it may not be achieved by any strategy for the 1-RMDP. By contrast, for BMDPs we show that if the supremum reachability probability is 1, then there is a strategy that achieves it. This is one illustration of the fact that the equivalence between 1-RMDP and BMDP does not hold for the reachability objective. The papers [4, 3] do not address the limit-sure reachability problem, and in fact even the decidability of limit-sure reachability for 1-RMDPs remains open.

Chen et. al. [5] studied model checking of branching processes with respect to properties expressed by deterministic parity tree automata and showed that the qualitative problem is in P (hence this holds in particular for reachability probability in BPs), and that the quantitative problem of comparing the probability with a rational is in PSPACE. Although not explicitly stated there, one can use Lemma 20 of [5] and our algorithm from [8] to show that the reachability probabilities of BPs can be approximated in P-time. Bonnet et. al. [2] studied a model of "probabilistic Basic Parallel Processes", which are syntactically close to Branching processes, except reproduction is asynchronous and the entity that reproduces in each step is chosen randomly (or by a scheduler/controller). None of the previous results have direct bearing on the reachability problems for BMDPs.

Due to space limits, most proofs are omitted. See the full version [10].

2 Definitions and Background

We provide unified definitions of multi-type Branching processes (BPs), Branching MDPs (BMDPs), and Branching Simple Stochastic Games (BSSGs), by first defining BSSGs, and then specializing them to obtain BMDPs and BPs.

A *Branching Simple Stochastic Game* (BSSG), consists of a finite set $V = \{T_1, \ldots, T_n\}$ of types, a finite non-empty set $A_i \subseteq \Sigma$ of actions for each type (Σ is some finite action alphabet), and a finite set $R(T_i, a)$ of probabilistic rules associated with each pair (T_i, a), $i \in [n]$, where $a \in A_i$. Each rule $r \in R(T_i, a)$ is a triple (T_i, p_r, α_r), which we denote by $T_i \xrightarrow{p_r} \alpha_r$, where $\alpha_r \subseteq \mathbb{N}^n$ is a n-vector of natural numbers that denotes a finite multi-set over the set V, and where $p_r \in (0, 1]$ is the probability of the rule r, where $\sum_{r \in R(T_i,a)} p_r = 1$ for all $i \in [n]$ and $a \in A_i$. For BSSGs, the types are partitioned into two sets: $V = V_{\max} \cup V_{\min}$, $V_{\max} \cap V_{\min} = \emptyset$, where $V_{\max}$ contains those types "belonging" to player max, and $V_{\min}$ containing those belonging to player min. A *Branching Markov Decision Process* (BMDP) is a BSSG where one of the two sets $V_{\max}$ or $V_{\min}$ is empty. Intuitively, a BMDP (BSSG) describes the stochastic evolution of a population of entities of different types in the presence of a controller (or two players) that can influence the evolution. A *multi-type Branching Process* (BP), is a BSSG where all action sets A_i are singleton sets; hence in a BP players have no choices and thus don't exist: a BP defines a purely stochastic process.

A play (or trajectory) of a BSSG operates as follows: starting from an initial population (i.e., set of entities of given types) X_0 at time (generation) 0, a sequence of populations $X_1, X_2, \ldots$ is generated, where X_{k+1} is obtained from X_k as follows. Player max (min) selects for each entity e in set X_k that belongs to max (to min, respectively) an available action $a \in A_i$ for the type T_i of entity e; then for each such entity e in X_k a rule $r \in R(T_i, a)$ is chosen randomly and independently according to the rule probabilities p_r, where $a \in A_i$ is the action selected for that particular entity e. Every entity is then replaced by a set of entities with the types specified by the right-hand side multiset α_r of that chosen rule r. The process is repeated as long as the current population X_k is nonempty, and it is said to *terminate* (or become *extinct*) if there is some $k \geq 0$ such that $X_k = \emptyset$. When there are n types, we can view a population X_i as a vector $X_i \in \mathbb{N}^n$, specifying the number of objects of each type. We say that the process *reaches* a type T_j, if there is some $k \geq 0$ such that $(X_k)_j > 0$.

A player can base her decisions at each stage k on the entire past history, and may choose different actions for entities of the same type.[3] The decision may be *randomized* (i.e. a probability distribution on the tuples of actions for the entities of the types controlled by the player) or *deterministic* (see the full version [10] for the formal definitions). Let Ψ_1, Ψ_2 be the set of all (randomized) strategies of the two players. We say that a strategy is *static* if for each type T_i

[3] We remark that, for optimizing termination and reachability probability, we could alternatively define the players' strategic role in BSSGs in various other ways, including asynchronous choice of (randomized) actions, as long as actions must eventually be chosen for all objects, without altering any of our results.

controlled by that player the strategy always chooses the same action a_i, or the same probability distribution on actions, for all entities of type T_i in all histories.

We can consider different objectives by the players. Here we consider the *reachability* objective, where the goal of the two players, starting from a given population, is to maximize/minimize the probability of reaching a population which contains *at least* one entity of a given special type, T_{f^*}. It will follow from our results that a BSSG game with a reachability objective has a *value*.

Suppose that player 1 wants to maximize the probability of *not* reaching T_{f^*} and player 2 wants to minimize it. For strategies $\sigma \in \Psi_1$, $\tau \in \Psi_2$, and a given initial population $\mu \in \mathbb{N}^n$, with $(\mu)_{f^*} = 0$, we denote by $g^{*,\sigma,\tau}(\mu)$ the probability that $(X_d)_{f^*} = 0$ for all $d \geq 0$. The *value* of the non-reachability game for the initial population μ is $g^*(\mu) = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} g^{*,\sigma,\tau}(\mu)$. We will show that determinacy holds for these games, i.e., $g^*(\mu) = \sup_{\sigma\in\Psi_1} \inf_{\tau\in\Psi_2} g^{*,\sigma,\tau}(\mu) = \inf_{\tau\in\Psi_2} \sup_{*,\sigma\in\Psi_1} g^{*,\sigma,\tau}(\mu)$. However, unlike the case for extinction probabilities ([12]), it does *not* hold that both players have optimal static strategies.

If μ has a single entity of type T_i, we will write g_i^* instead of $g^*(\mu)$. Given a BMDP (or BSSG), the goal is to compute the vector g^* of the g_i^*'s, i.e. the vector of non-reachability values of the different types. From the g_i^*'s, we can compute the value $g^*(\mu)$ for any initial population μ: $g^*(\mu) = \Pi_i (g_i^*)^{\mu_i}$.

We will associate a system of min/max probabilistic polynomial Bellman equations, $x = P(x)$, to each given BMDP or BSSG. A polynomial $p(x)$ is called *probabilistic* if all coefficients are nonnegative and sum to at most 1. A *probabilistic polynomial system (PPS)* is a system $x = P(x)$ where all $P_i(x)$ are probabilistic polynomials. A *max-min PPS* is a system $x = P(x)$ where each $P_i(x)$ is either: a `Max-polynomial`: $P_i(x) = \max\{q_{i,j}(x) : j \in \{1, ..., m_i\}\}$, or a `Min-polynomial`: $P_i(x) = \min\{q_{i,j}(x) : j \in \{1, ..., m_i\}\}$, where each $q_{i,j}(x)$ is a probabilistic polynomial, for every $j \in \{1, \ldots, m_i\}$. We shall call such a system a *maxPPS* (respectively, a *minPPS*) if for every $i \in \{1, \ldots, n\}$, $P_i(x)$ is a Max-polynomial (respectively, a Min-polynomial). We use *max/minPPS* to refer to a system of equations, $x = P(x)$, that is either a maxPPS or a minPPS.

For computational purposes we assume that all coefficients are rational, and that the polynomials are given in sparse form, i.e., by listing only the nonzero terms, with the coefficient and the nonzero exponents of each term given in binary. We let $|P|$ denote the total bit encoding length of a system $x = P(x)$ under this representation.

Any max-minPPS, $x = P(x)$, has a *least fixed point* (**LFP**) solution, $q^* \in [0,1]^n$, i.e., $q^* = P(q^*)$ and if $q = P(q)$ for some $q \in [0,1]^n$ then $q^* \leq q$ (coordinate-wise inequality). As observed in [13, 12], q^* may in general contain irrational values, even in the case of pure PPSs. In this paper, we exploit the fact that every max-minPPS, $x = P(x)$, also has a *greatest fixed point* (**GFP**) solution, $g^* \in [0,1]^n$, i.e., such that $g^* = P(g^*)$ and if $q = P(q)$ for some $q \in [0,1]^n$ then $q \leq g^*$. Again, g^* may contain irrational coordinates, so we in general want to approximate its coordinates.

We can consider a max-minPPS as a game between two players that control respectively the variables x_i where P_i is a max or a min polynomial. A

(possibly randomized) policy σ for a player maps each of its variables x_i to a probability distribution $\sigma(i)$ over the indices $\{1, \ldots, m_i\}$ of the polynomials in P_i. A policy σ of the max player induces a minPPS $x = P_\sigma(x)$, where $(P_\sigma)_i(x) = \sum_{a \in A_i} \sigma(i)(a) \cdot q_{i,a}$. Let q^*_σ and g^*_σ denote the LFP and GFP of the min-PPS $x = P_\sigma(x)$. We say that σ is an *optimal* policy for the max player for the LFP (resp., the GFP) if $q^*_{\sigma^*} = q^*$ (resp., $g^*_{\sigma^*} = g^*$). The policy σ is ϵ-*optimal* for the LFP (resp. GFP) , if $||q^*_{\sigma'} - q^*||_\infty \leq \epsilon$ (resp., $||g^*_{\sigma'} - g^*||_\infty \leq \epsilon$). These concepts can be defined similarly for the min player and its policies.

It is convenient to put max-minPPSs in the following simple form.

Definition 1. *A max-minPPS, $x = P(x)$ in n variables is in* simple normal form (SNF) *if each $P_i(x)$, for all $i \in [n]$, is in one of the following three forms:*
Form `L`*: $P(x)_i = a_{i,0} + \sum_{j=1}^{n} a_{i,j} x_j$, where $a_{i,j} \geq 0$ for all j, & $\sum_{j=0}^{n} a_{i,j} \leq 1$.*
Form `Q`*: $P(x)_i = x_j x_k$ for some j, k.*
Form `M`*: $P(x)_i = \max\{x_j, x_k\}$ or $P(x)_i = \min\{x_j, x_k\}$, for some j, k.*

We define SNF form for max/minPPSs analogously. Every max-minPPS, $x = P(x)$, can be transformed in P-time (as in [8, 13]) to a suitably "*equivalent*" max-minPPS in SNF form (see the full version [10] for a formal statement and proof), where in particular both the LFP *and* GFP of the original system are projections of the LFP and GFP of the transformed systems. Thus *we may (and do) assume*, wlog, that all max/minPPSs are in SNF normal form.

The *dependency graph* of a max-minPPS $x = P(x)$ is a directed graph with one node for each variable x_i, and contains edge (x_i, x_j) iff x_j appears in $P_i(x)$.

For a max/minPPS, $x = P(x)$, with n variables (in SNF form), the *linearization of $P(x)$ at a point* $\mathbf{y} \in \mathbb{R}^n$, is a system of max/min linear functions denoted by $P^y(x)$, which has the following form: if $P(x)_i$ has form `L` or `M`, then $P^y_i(x) = P_i(x)$, and if $P(x)_i$ has form `Q`, i.e., $P(x)_i = x_j x_k$ for some j, k, then $P^y_i(x) = y_j x_k + x_j y_k - y_j y_k$. We now recall and adapt from [9] the definition of distinct iteration operators for maxPPSs and minPPSs, both of which we shall refer to with the overloaded notation $I(x)$. These operators serve as the basis for *Generalized Newton's Method* (GNM) to be applied to maxPPSs and minPPSs, respectively. We need to slightly adapt the definition of operator $I(x)$, specifying the conditions on the GFP g^* under which the operator is well-defined:

Definition 2. *For a maxPPS, $x = P(x)$, with GFP g^*, with $0 \leq g^* < 1$, and for $0 \leq y \leq g^*$, define the operator $I(y)$ to be the* unique *optimal solution, $a \in \mathbb{R}^n$, to the following mathematical program:* Minimize: $\sum_i a_i$; Subject to: $P^y(a) \leq a$.

For a minPPS, $x = P(x)$, with GFP g^*, with $0 \leq g^* < 1$, and for $0 \leq y \leq g^*$, define the operator $I(y)$ to be the unique *optimal solution $a \in \mathbb{R}^n$ to the following mathematical program:* Maximize: $\sum_i a_i$; Subject to: $P^y(a) \geq a$.

These mathematical programs can be solved using Linear Programming. A priori, it is unclear whether the programs have a unique solution, i.e., whether the "definitions" of $I(x)$ for maxPPSs and minPPSs are well-defined. We show they are. We require *rounded* GNM, defined as follows ([9]).

GNM, with rounding parameter h: Starting at $x^{(0)} := \mathbf{0}$, For $k \geq 0$, compute $x^{(k+1)}$ from $x^{(k)}$ as follows: first calculate $I(x^{(k)})$, then for every coordinate i, set $x_i^{(k+1)}$ to be the maximum multiple of 2^{-h} which is $\leq \max\{0, I(x^{(k)})_i\}$.

3 Greatest Fixed Points capture non-reachability values

For any given BSSG, $\mathcal{G}$, with a specified special type T_{f^*}, we will construct a max-minPPS, $x = P(x)$, and show that the vector g^* of *non*-reachability values for $(\mathcal{G}, T_{f^*})$ is precisely the *greatest fixed point* $g^* \in [0,1]^n$ of $x = P(x)$.

The system $x = P(x)$ has one variable x_i and one equation $x_i = P_i(x)$, for each type $T_i \neq T_{f^*}$. For each $i \neq f^*$, the min/max probabilistic polynomial $P_i(x)$ is constructed as follows. For all $j \in A_i$, let $R'(T_i, j) := \{r \in R(T_i, j) : (\alpha_r)_{f^*} = 0\}$ denote the set of rules for type T_i and action j that generate a multiset α_r not containing any element of type T_{f^*}. $P_i(x)$ contains one probabilistic polynomial $q_{i,j}(x)$ for each action $j \in A_i$, with $q_{i,j}(x) = \sum_{r \in R'(T_i,j)} p_r x^{\alpha_r}$. Note that we *do not* include, in the sum defining $q_{i,j}(x)$, any monomial $p_{r'} x^{\alpha_{r'}}$ associated with a rule r' which generates an object of the special type T_{f^*}. Then, if type T_i belongs to player max, who aims to *minimize* the probability of *not* reaching an object of type T_{f^*}, we define $P_i(x) \equiv \min_{j \in A_i} q_{i,j}(x)$. Likewise, if T_i belongs to min, whose aim is to *maximize* the probability of *not* reaching T_{f^*}, we define $P_i(x) \equiv \max_{j \in A_i} q_{i,j}(x)$. Note the swapped roles of max and min in the equations, versus the corresponding player's goal for the reachability objective. The following theorem is analogous to one in [12] for LFPs of max-minPPSs.

Theorem 1. *The value vector $g^* \in [0,1]^n$ of a BSSG is the GFP of the corresponding operator $P(\cdot)$ in $[0,1]^n$. Thus, $g^* = P(g^*)$, and $\forall g' \in [0,1]^n$, $g' = P(g')$ implies $g' \leq g^*$. Also, for any initial population μ, the non-reachability values satisfy $g^*(\mu) = \sup_{\sigma \in \Psi_1} \inf_{\tau \in \Psi_2} g^{*,\sigma,\tau}(\mu) = \inf_{\tau \in \Psi_2} \sup_{\sigma \in \Psi_1} g^{*,\sigma,\tau}(\mu) = \Pi_i (g_i^*)^{\mu_i}$. So, such games are* determined.

A direct corollary of the proof of Theorem 1 (see the full version [10]) is that the player maximizing non-reachability probability in a BSSG always has an optimal deterministic static strategy. The same is *not* true for the player trying to *minimize* this non-reachability probability (i.e. the player trying to maximize the reachability probability). We give two examples illustrating this (see [10] for details). The first example has types A, B, and C, start type A and target type B, only A is controlled; B is purely probabilistic. The rules are: $A \to AA$, $A \to B$, $B \stackrel{1/2}{\to} C$, $B \stackrel{1/2}{\to} \emptyset$. There is no randomized static optimal strategy for maximizing the reachability probability in this BMDP, although the supremum probability is 1. We show later however that for any BMDP, if the supremum reachability value is 1, then the player maximizing the reachability probability has a, not necessarily static, optimal strategy that achieves value 1. The second example shows that this is not the case if the value is strictly between 0 and 1. Consider the BMDP with types A, B, C, and D, start type A and target type D, with rules: $A \stackrel{2/3}{\to} BB$, $A \stackrel{1/3}{\to} \emptyset$, $B \to A$, $B \to C$, $C \stackrel{1/3}{\to} D$, $C \stackrel{2/3}{\to} \emptyset$. There is no optimal strategy for maximizing the reachability probability in this BMDP (i.e., the supremum, which is 1/2, is not achievable by any strategy), see [10].

Qualitative $= 1$ non-reachability analysis for BSSGs & max-minPPSs. There are (easy) P-time algorithms to compute for a given max-minPPS the

variables that have value 1 in the GFP, and thus also for deciding, for a given BSSG (or BMDP), whether $g_i^* = 1$ (i.e., whether the *non*-reachability value is 1). The easy algorithm boils down to AND-OR graph reachability.

Proposition 1. *There is a P-time algorithm that given a max-min-PPS, $x = P(x)$, with n variables, and with GFP $g^* \in [0,1]^n$, and given $i \in [n]$, decides whether $g_i^* = 1$, or $g_i^* < 1$; Moreover, when $g_i^* = 1$ the algorithm outputs a deterministic policy (i.e., deterministic static strategy for the BSSG) σ, for the max player which forces $g_i^* = 1$, Likewise, if $g_i^* < 1$, it outputs a deterministic static policy τ for the min player which forces $g_i^* < 1$.*

We consider detection of $g_i^* = 0$ for maxPPS and minPPS later; the minPPS case in particular is substantially more complicated.

4 maxPPSs

We first determine and remove the variables with value 1 in the GFP, after which we know $g^* < 1$. To analyze maxPPSs, we first perform a thorough structural analysis of PPSs (without max) and derive several properties that are useful in handling maxPPSs (and minPPSs). Building on these properties, we show:

Lemma 1. *For any maxPPS, $x = P(x)$, if GFP $g^* < 1$ then g^* is the unique fixed point of $x = P(x)$ in $[0,1]^n$. So $g^* = q^*$, where q^* is the LFP of $x = P(x)$.*

Thus, applying the algorithms from [9] for LFP computation of maxPPSs, yields:

Theorem 2. *Given a maxPPS, $x = P(x)$, with GFP g^*,*

1. *There is a P-time algorithm that determines, for $i \in [n]$, whether $g_i^* = 0$, and if $g_i^* > 0$ computes a deterministic static policy that achieves this.*
2. *Given any integer $j > 0$, there is an algorithm that computes a rational vector v with $\|g^* - v\|_\infty \leq 2^{-j}$, and also computes a deterministic static policy σ, such that $\|g^* - g_\sigma^*\| \leq 2^{-j}$, both in time polynomial in $|P|$ and j.*

Similar results follow for the maximization of nonreachability in BMDPs.

5 minPPSs

Theorem 3. *Given a minPPS, $x = P(x)$ with $g^* < 1$. If we use GNM with rounding parameter $h = j + 2 + 4|P|$, then after h iterations, we have $\|g^* - x^{(h)}\|_\infty \leq 2^{-j}$. This ϵ-approximates g^* in time polynomial in $|P|$ and $\log(\frac{1}{\epsilon})$.*

The minPPS case is much more involved. In order to prove this theorem, we need some structural lemmas about GFPs of minPPSs, and their relationship to static policies. There need not exist any policies σ with $g_\sigma^* = g^*$, so we need policies that can, in some sense, act as "surrogates" for it. We say that a PPS $x = P(x)$ is *linear degenerate* (LD) if every $P_i(x)$ is a convex combination of variables: $P_i(x) \equiv \sum_{j=1}^n p_{ij} x_j$ where $\sum_j p_{ij} = 1$. A PPS is *linear degenerate free (LDF)* if there is no bottom strongly connected component S of its dependency

graph, whose induced subsystem $x_S = P_S(x_S)$ is linear degenerate. A policy σ for a max/minPPS, $x = P(x)$, is called linear degenerate free (LDF) if its associated PPS $x = P_\sigma(x)$ is an LDF PPS. It turns out there is an LDF policy σ^* whose associated LFP is the GFP of the minPPS, and we can get an ϵ-optimal policy by following σ^* with high probability and with low probability following some policy that can reach the target from anywhere.

Lemma 2. *If a minPPS $x = P(x)$ has $g^* < 1$ then:*

1. *There is an LDF policy σ with $g^*_\sigma < 1$,*
2. *$g^* \leq q^*_\tau$, for any LDF policy τ, and*
3. *There is an LDF policy σ^* whose associated LFP, $q^*_{\sigma^*}$, has $g^* = q^*_{\sigma^*}$.*

Note that the policy σ^* is not necessarily optimal because even though $g^* = q^*_{\sigma^*}$, there may be an i with $g^*_i = (q^*_{\sigma^*})_i < (g^*_{\sigma^*})_i = 1$. Next we show that Generalised Newton's Method (GNM) is well-defined. We use $\mathcal{N}_\sigma$ below to denote the standard Newton iteration operator applied to the PPS $x = P_\sigma(x)$ (see [10]).

Lemma 3. *Given a minPPS, $x = P(x)$, with GFP $g^* < 1$, and given y with $0 \leq y \leq g^*$, there exists an LDF policy σ with $P^y(\mathcal{N}_\sigma(y)) = \mathcal{N}_\sigma(y)$, the GNM operator $I(x)$ is defined at y, and for this policy σ, $I(y) = \mathcal{N}_\sigma(y)$.*

Using this, we can show a result for GFPs similar to one in [9] for LFPs:

Lemma 4. *Let $x = P(x)$ be a minPPS with GFP $g^* < 1$. For any $0 \leq x \leq g^*$ and $\lambda > 0$, $I(x) \leq g^*$, and if $g^* - x \leq \lambda(1 - g^*)$ then $g^* - I(x) \leq \frac{\lambda}{2}(1 - g^*)$.*

Theorem 3 follows by using Lemma 4 and Lemma 2(3.), and applying a similar inductive argument as in ([9], Section 3.5).

P-time detection of zeros in the GFP of a minPPS: $g^*_i \stackrel{?}{=} 0$.

We give a P-time algorithm for deciding whether the supremum reachability probability in a BMDP equals 1, in which case we show the supremum probability is achieved by a (memoryful but deterministic) strategy which we can compute in P-time (thus limit-sure and almost-sure reachability are the same). Let X be the set of all variables x_i in minPPS $x = P(x)$ in SNF form, with GFP $g^* < 1$.

1. Initialize $S := \{\ x_i \in X \mid P_i(0) > 0$, i.e., $P_i(x)$ contains a constant term $\}$.
2. Repeat the following until neither are applicable:
   (a) If a variable x_i is of form `L` and $P_i(x)$ has a term whose variable is already in S, then add x_i to S.
   (b) If a variable x_i is of form `Q` or `M` and both variables in $P_i(x)$ are already in S, then add x_i to S.
3. Let $F := \{\ x_i \in X - S \mid P_i(1) < 1$, or $P_i(x)$ has form `Q` $\}$.
4. Repeat the following until no more variables can be added:
   – If a variable $x_i \in X - S$ is of form `L` or `M` and P_i contains a term whose variable is in F, then add x_i to F.
5. If $X = S \cup F$, then terminate and output F.
6. Otherwise set $S := X - F$ and return to step 2.

Theorem 4. *Given a minPPS $x = P(x)$ with $g^* < 1$, this algorithm terminates and outputs precisely the variables x_i with $g^*_i = 0$, in time polynomial in $|P|$.*

Theorem 5. *There is a non-static deterministic optimal strategy for maximizing the probability of reaching a target type in a BMDP with probability 1, if the supremum probability of reaching the target is 1.*

We outline the non-static policy. The proof of Theorem 4 constructs a LDF policy σ with the property that $g^*_i = 0$ iff $(q^*_\sigma)_i = 0$. Let Z denote the set of variables with $g^*_i = 0 = (q^*_\sigma)_i$. From Proposition 1, we can also compute in P-time an LDF policy τ with $g^*_\tau < 1$. We combine σ and τ in the following non-static policy: We designate one member of our initial population with type in Z to be the queen. The rest of the population are workers. We use policy σ for the queen and τ for the workers. In following generations, if we have not reached an object of the target type, we choose one of the children in Z of the last generation's queen (which we show must exist) to be the new queen. Again, all other members of the population are workers.

Computing ϵ-optimal strategies for minPPSs in P-time:

We first use the following algorithm to find an LDF policy σ with $\|g^* - q^*_\sigma\|_\infty \leq \frac{1}{2}\epsilon$. We then use that policy to construct ϵ-optimal policies.

1. Compute, using GNM, a $0 \leq y \leq g^*$ with $\|g^* - y\|_\infty \leq 2^{-14|P|-3}\epsilon$;
2. Let $k := 0$, and let σ_0 be a policy that has $P_{\sigma_0}(y) = P(y)$ (i.e., σ_0 chooses the action with highest probability of reaching the target according to y).
3. Compute F_{σ_k}, the set of variables that, in the dependency graph of $x = P_{\sigma_k}(x)$, either are or depend on a variable x_i which either has form `Q` or else $P_i(\mathbf{1}) < \mathbf{1}$ or $P_i(0) > 0$. Let D_{σ_k} be the complement of F_{σ_k} .
4. if D_{σ_k} is empty, we are done, and we output σ_k.
5. Find a variable[4] x_i of type `M` in D_{σ_k}, which has a choice x_j in F_{σ_k} (which isn't its current choice) such that $|y_i - y_j| \leq 2^{-14|P|-2}\epsilon$; Let policy σ_{k+1} choose x_j at x_i, & otherwise agree with σ_k. Let $k := k+1$; return to step 3.

Lemma 5. *The above algorithm terminates in P-time and outputs an LDF policy σ with $\|P_\sigma(y) - y\|_\infty \leq 2^{-14|P|-2}\epsilon$.*

We define a randomized static policy υ as follows. With probability $2^{-28|P|-4}\epsilon$ we follow a (necessarily LDF) deterministic policy τ that satisfies $g^*_\tau < 1$. We can compute such a τ in P-time by Proposition 1. With the remaining probability $1 - 2^{-28|P|-4}\epsilon$, we follow the static deterministic policy σ that is output by the algorithm above. We can then show (see [10] for the involved proof):

Theorem 6. *The output policy σ of the algorithm satisfies $\|g^* - q^*_\sigma\|_\infty \leq \frac{1}{2}\epsilon$. Moreover, υ satisfies $\|g^* - g^*_\upsilon\|_\infty \leq \epsilon$, i.e., it is ϵ-optimal.*

[4] We show that such a variable x_i always exists whenever we reach this step.

Theorem 7. *For a BMDP with minPPS $x = P(x)$, and minimum non-reachability probabilities given by the GFP $g^* < 1$, the following deterministic non-static non-memoryless strategy α is also ϵ-optimal starting with one object of any type: Use the policy σ output by the algorithm, until the population is at least $\frac{2^{4|P|+1}}{\epsilon}$ for the first time, thereafter use a deterministic static policy τ such that $g^*_\tau < 1$.*

Corollary 1. *For maximizing BMDP reachability probability, we can compute in P-time a randomized static (or deterministic non-static) ϵ-optimal policy.*

References

[1] Bozic, et. al. Evolutionary dynamics of cancer in response to targeted combination therapy. *Elife*, volume 2, pages e00747, 2013.

[2] R. Bonnet, S. Kiefer, A. W. Lin: Analysis of Probabilistic Basic Parallel Processes. In *Proc. of FoSSaCS'14*, pages 43-57, 2014.

[3] T. Brázdil, V. Brozek, A. Kucera, J. Obdrzálek: Qualitative reachability in stochastic BPA games. *Inf. Comput.*, 209(8): 1160-1183, 2011.

[4] T. Brázdil, V. Brozek, V. Forejt, and A. Kucera. Reachability in recursive Markov decision processes. *Inf. Comput.*, 206(5):520–537, 2008.

[5] T. Chen, K. Dräger, S. Kiefer: Model checking stochastic branching processes. In Proc. of *MFCS'12*, Springer LNCS 7464, pages 271-282, 2012.

[6] J. Esparza, T. Gawlitza, S. Kiefer, and H. Seidl. Approximative methods for monotone systems of min-max-polynomial equations. In *Proc. 35th ICALP*, 2008.

[7] J. Esparza, A. Kučera, and R. Mayr. Model checking probabilistic pushdown automata. *Logical Methods in Computer Science*, 2(1):1 – 31, 2006.

[8] K. Etessami, A. Stewart, and M. Yannakakis. Polynomial-time algorithms for multi-type branching processes and stochastic context-free grammars. In *Proc. 44th ACM Symposium on Theory of Computing (STOC)*, 2012.

[9] K. Etessami, A. Stewart, and M. Yannakakis. Polynomial-time algorithms for Branching Markov Decision Processes, and probabilistic min(max) polynomial Bellman equations. In *Proc. 39th ICALP*, 2012. (Full preprint: ArXiv:1202.4789).

[10] Full preprint of this paper: arXiv:1502.05533 (2015).

[11] K. Etessami, D Wojtczak, and M. Yannakakis. Recursive stochastic games with positive rewards. In *Proc. of 35th ICALP (1)*, pages 711–723. Springer, 2008.

[12] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. *Journal of the ACM*, 2015.

[13] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. *Journal of the ACM*, 56(1), 2009.

[14] S. Pliska. Optimization of multitype branching processes. *Management Sci.*, 23(2):117–124, 1976/77.

[15] G. Reiter, I. Bozic, K. Chatterjee, M. A. Nowak. TTP: Tool for tumor progression. In *Proc. of CAV'2013*, pages 101-106, Springer LNCS 8044, 2013.

[16] U. Rothblum and P. Whittle. Growth optimality for branching Markov decision chains. *Math. Oper. Res.*, 7(4):582–601, 1982.